

UE4 管理多重项目-WiseeEngineering

作者 : Joe Wilcox

介绍

在 WisEngineering , 我们随时都有多个处于不同生产状态的 UnrealEngine 4 (UE4) 项目。尽管 UE4 是大小团队的出色解决方案, 但管理 UE4 项目可能会令人生畏。大型团队将从 Unreal Game Sync (UGS) 系统的大量资源中受益, 从而为构建版本分配提供了很好的解决方案。但是, 对于较小的组或对远程访问有深入要求的组, UGS 并非易于实施的解决方案。本白皮书适合您。

本文将阐明管理构建并将其分发给团队成员的过程, 为艺术家和程序员提供一条路径, 并演示如何设置最佳的解决方案以进行合并和合并。

升级。

关于作者

Joe Wilcox 是 WisEngineering 的技术总监, 通过 20 多年在虚幻引擎中构建游戏软件的经验, 并将 WisE 定位于 AR / VR 革命的最前沿, 并在 6 个平台上交付了 15 多种游戏。

不同类型的 UE4 发行版

在开始剖析如何与小型团队一起管理项目之前, 重要的是要了解所有三种用于分发构建的现有方法。Epic 在[此处](#)对各种部署方法进行了基本描述。让我们更详细地细分它们。

1.检查内置到源代码管理

这是分发内部版本的原始方法, 在我开发 UT Alpha 时在 Epic 中使用。从开发的角度来看, 将内部版本检查到源代码管理很简单, 程序员或开发机器创建内部版本并检入新创建的二进制文件。

内容开发人员会为其平台下载预编译的内部版本，并利用该内部版本来满足其工作负载。程序员通常会重新编译自己的项目本地构建。构建标签通常用于更好地控制团队成员使用的构建。听起来不错吧？乍一看，perforce 的构建很简单，但是它们也有一些缺点。首先，UE4 构建的大小使它们难以传递。如果您经常检查项目和引擎的二进制文件，那么您将面临大量同步时间和庞大文件传输的麻烦。

其次，由于每个项目都有其本地来源和完整的引擎来源，因此升级到较新的引擎版本可能是一个漫长的过程。开发人员必须单独升级每个项目。

最后，通过使用源代码控制来分发二进制文件，通常可以在组合中获得毒药，即破坏游戏内容的构建。

2.利用虚幻游戏同步系统

UGS 系统是一种管理源代码控制中构建的有效方法，它可以减轻许多负面影响。它还需要在后端资源上进行大量投资，以确保可以正确跟踪构建并将其分发给最终用户。话虽如此，它简化了对升级版本的管理，并有助于防止生成有害代码。

将 UGS 用作小型开发公司的最大弊端是执行管理所需的后端基础结构。

如果您想了解有关 UGS 的更多信息，建议您从以下链接开始：

[UnrealGameSync 文档](#)

3.通过自定义的已安装版本进行分发

自定义安装的版本就像使用 Epic Games Store 安装引擎的版本并从中进行工作一样。引擎的主要完整安装将以类似于精简安装版本的方式进行处理。然后可以将此构建检查到源代码管理中或以某种方式分发。

使用引擎的自定义安装版本有几个优点，但是也有一些严重的限制。这种方法的优点是，您的所有项目都基于引擎的固定开发版本。这意味着升级到较新的发动机卸货是一个轻松而又快速的过程。但是，这也强调了最大的缺点。自定义安装版本仅适用于需要对基本引擎进行最小更改并且这些更改要在所有项目之间共享的情况。

您选择的方法应取决于您的情况。在 WisEngineering，我们有许多正在为多个客户进行积极开发的项目，但是我们对引擎所做的更改很小，而且

在所有项目中共享。因此，自定义构建的紧密集成和更快的编译时间适合我们的开发风格。本白皮书将布局我们的安装，维护和利用自定义安装版本以供多个项目使用的过程。让我们开始。

新术语

在进入建立自定义安装构建系统的本质之前，我们应该确保我们在术语和目录布局方面处于同一页面。这将在以后变得很重要。

Custom Install Build —一种利用预编译的 UE4 引擎二进制文件进行开发的方法。

Source Drop —这是某种形式的引擎副本。有两种类型的 Source Drop：引擎代码 Drop 和开发代码 Drop。

Engine drop —在这里，您可以保留用于创建安装版本的基本引擎源代码。该系统中只能有一个单独的引擎 Drop。

Development drop —这是引擎的内部版本，已打包到自定义安装版本中。如果我们需要将项目锁定到特定的引擎分支，我们的系统支持多个开发分支。通常，将有一个引擎下降而一个开发下降。

目录结构

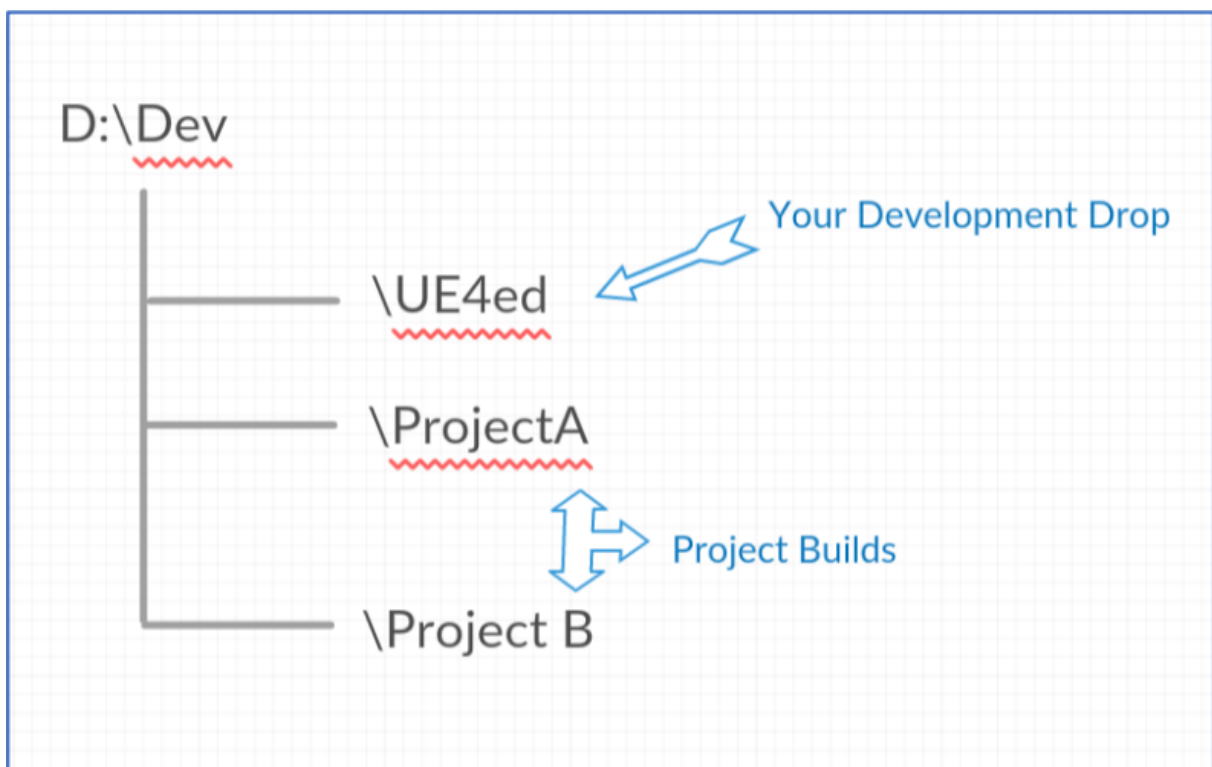


图 1.1 –程序员目录结构

图 1.1 显示了典型的程序员目录结构。 WisE 的程序员可以访问 Engine 下拉列表和 Development 下拉列表，以方便更改我们的引擎插件。 但是，只有负责创建安装版本的人员才真正需要访问引擎投递。

我们的艺术家具有如图 1.2 所示的目录结构：

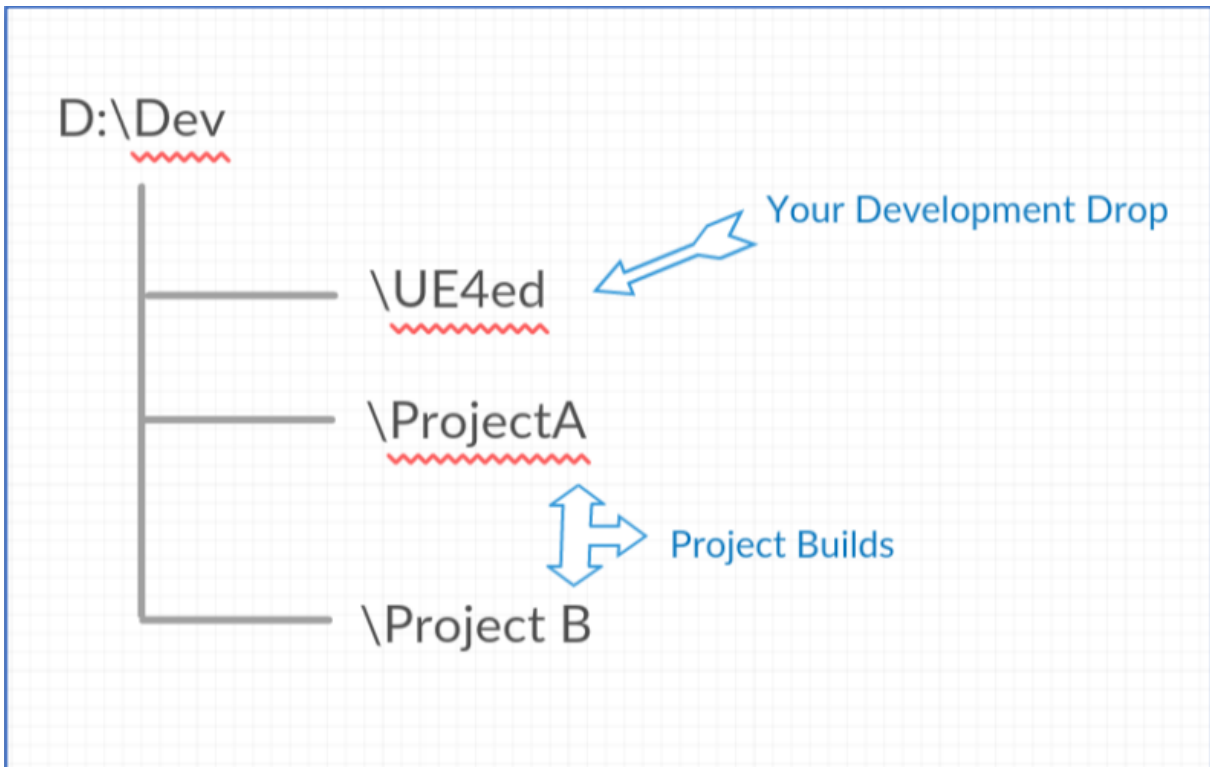


图 1.2 –艺术家目录结构

在这里，缺少引擎降落。 艺术家，一些程序员和蓝图脚本编写者不需要访问引擎放置，因此我们使用源代码控制角色来限制访问。 此目录结构非常适合管理项目源。 您还可以利用它来存储常用目录甚至示例代码。

从长远来看，尽早建立有效的目录结构将获得回报。 这种结构的好处之一是，它使在 Perforce 中使用流变得简单。

注意：像这样将项目存储在引擎目录之外需要使用完全限定的路径加载 UPROJECT 文件时。下面，我将展示如何在引擎级别解决此问题。

从头开始

那么我们如何设置环境？从 Epic Game 的 GitHub 页面下载源代码下载就可以了。您可以在

[UE4 GitHub](#) 页面上找到源。获取最新源代码的最简单方法是单击 GitHub 页面顶部的按钮

Clone or download ▾

。这将以 .ZIP 格式下载该引擎的最新批准版本。这是您的 **Engine drop**。

第 1 步-安装 Engine Drop

我们的下一步是安装该分支。您要在开发文件夹的根目录（本例中为 **D:\Dev**）中创建一个文件夹来保存引擎放置（本例中为 **D:\Dev\UE4**）。尽管您稍后需要记住该文件夹的位置，但您称其为“文件夹”并不重要。

步骤 2-安装必备组件

从 GITHUB .zip 文件解压缩所有文件后，您将需要安装所有必需的必备文件。史诗让这个超级简单。使用资源管理器浏览到引擎放置的根目录（在本例中为 **D:\Dev\UE4**），然后找到文件 **Setup.bat**。然后执行！

步骤 3 –初步编译

现在您的 PC 上已经安装了引擎 Drop，并且已经安装了所有必需的先决条件，接下来您需要进行第一次编译。首先，您需要生成您的项目文件。UE4 在根目录中有一个用于执行此操作的批处理文件（在我们的示例中为 **D:\Dev\UE4**）。浏览到该目录并运行

GenerateProjectFiles.bat。这将打开命令提示符并快速关闭。

现在，您可以通过打开将在同一目录中创建的 **UE4.sln** 来打开 Visual Studio 解决方案。

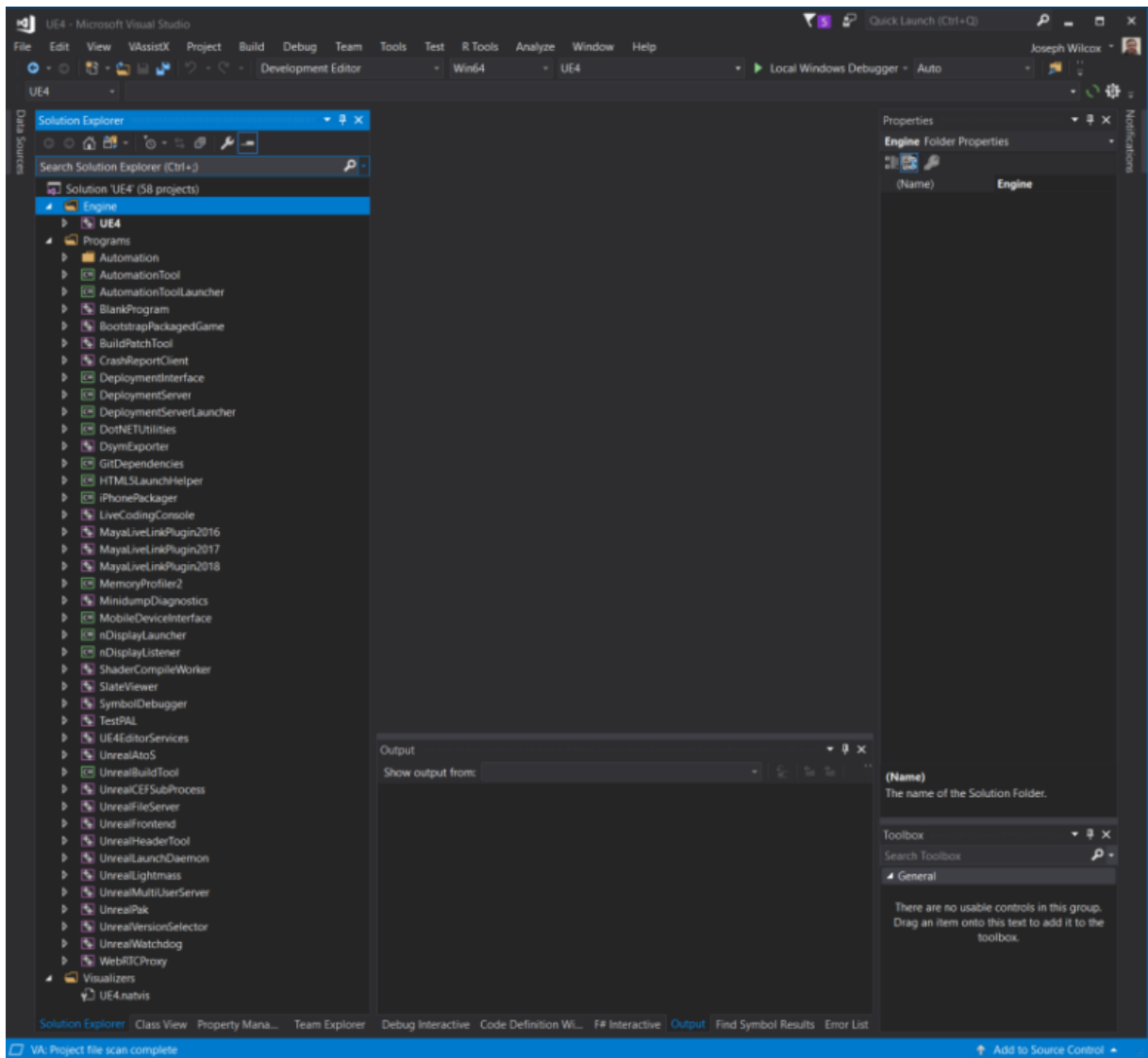


图 1.3—默认解决方案

加载 visualstudio 之后，您将需要展开 Engine 选项卡以将 UE4 项目显示为如图 1.4 所示：

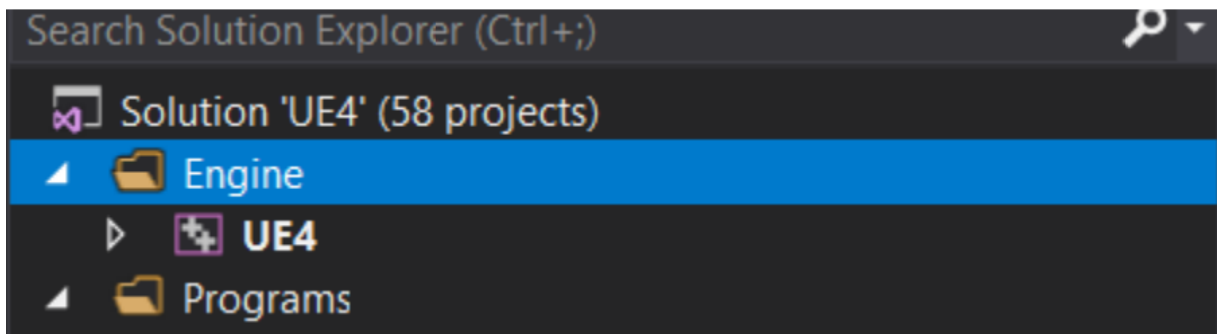


图 1.4-UE4 项目

右键单击 **UE4** 并选择 **Rebuild**。在 visualstudio 创建一组新的二进制 **binaries** 文件时，放松

一下，去喝一杯。

步骤 4—使 UE4 对多个项目更友好

一旦引擎的初始编译完成，就应该让 UE4 在多个项目中表现得更好。默认情况下，如果您使用的是自定义安装版本，则必须始终完全限定 `uproject` 文件的路径。虽然很小，但这种烦恼可以很快解决，使一个单一的引擎改变 `LaunchEngineLoop.cpp`，位于

`Engine\Source\Runtime\Launch\Private` 文件夹。

打开此文件并找到函数 `ParseGameProjectFromCommandLine()`。该函数的作用是获取命令行并从中提取 `uproject` 文件名。默认情况下，UE4 将查找任何文件夹在 UE4 中特定游戏 `uprojectdirs` 文件。但是，添加对父目录的引用 (`../`)不起作用。在内部 `If/Then/else` 块的 `else` 语句中，找到以下行：

```
// Full game name is assumed to be the first token
OutGameName = MoveTemp(FirstCommandLineToken);
// Derive the project path from the game name. All games must have a uproject file, even if they are
in the root folder.
OutProjectFilePath = FPaths::Combine(*FPaths::RootDir(), *OutGameName, *FString(OutGameName +
TEXT(".") + FProjectDescriptor::GetExtension()));
```

在这两行之后，我们要添加以下代码块：

```
if (!FPaths::FileExists(OutProjectFilePath))
{
    FString Root = FPaths::RootDir() + TEXT("../");
    FPaths::CollapseRelativeDirectories(Root);
    OutProjectFilePath = FPaths::Combine(*Root, *OutGameName,
    *FString(OutGameName + TEXT(".") +
    FProjectDescriptor::GetExtension()));
```

这看起来是为了确定 UE4 是否可以使用常规系统找到 UPROJECT 文件。

如果不能，请尝试返回到父目录，并使用 **UPROJECT** 文件名（不带扩展名）作为子目录名称，以建立 `uproject` 文件的完全限定路径。这与我在图 1.1 和图 1.2 中列出的目录结构配合使用非常好，并且可以节省很多输入时间。

注意：这是基本引擎的不受支持的更改。尽管我已经使用了一年多，但它可能仍会产生我尚未确定的副作用，尤其是在其他平台上。

步骤 5 –自动化工具

仅此工具我就可以写白皮书。现在，您将了解将使用 UE4 自动化工具来构建我们的自定义版本 UE4 的自定义安装版本。在进行此构建之前，您将必须对其进行编译，因为二进制文件并未创建为 UE4 项目的一部分。

在离开 Visual Studio 之前，请扩展解决方案的“程序”节点，并找到自动化工具项目，如图 1.5 所示。

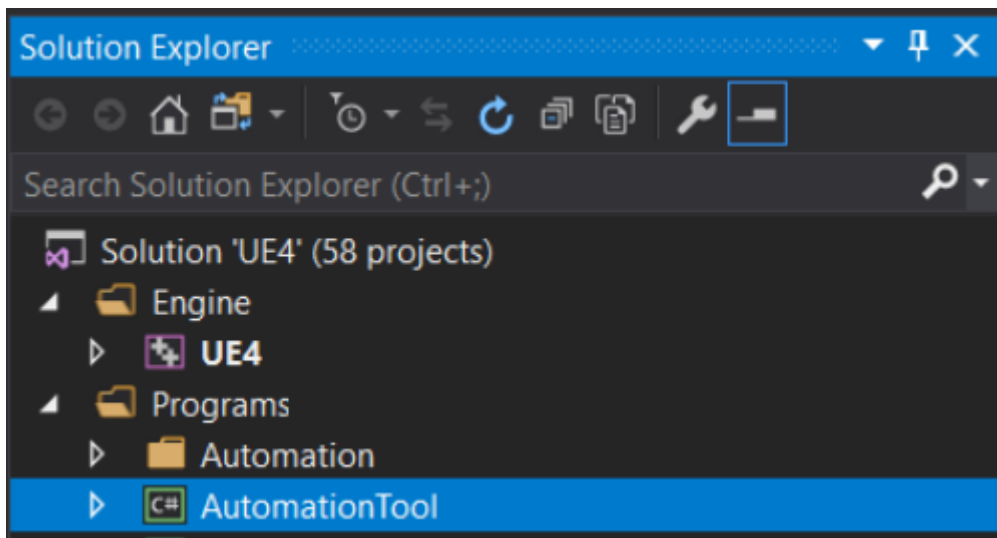


图 1.5 –自动化工具项目

再次右键单击该项目，然后选择“**Rebuild**（重建）”以创建自动化工具。与构建此引擎不同，与上面的完整重建相比，这将是一个快速编译。

我们的下一步是生成安装版本！

注意：在进行自定义安装之前，您需要直接安装所有引擎插件（如 Substance）或使用市场。随附的插件将包含在安装版本中。

生成安装版本

在继续执行步骤之前，重要的是要简要讨论一下 UE4 自动化工具（UAT），BuildGraph 和 InstallEngineBuild 脚本以及它们如何协同工作。UAT 是用于自动执行各种任务的特殊工具。这是一个复杂的系统，并且有许多脚本/用法。有关此系统的更多信息，建议您开始阅读《自动化系统概述》，因为我们将仅介绍该工具来生成我们的开发图。注意：在进行自定义安装之前，您需要直接安装所有引擎插件（即，Substance）或使用市场。随附的插件将包含

在

安装版本。

为了生成安装版本，我们将使用自定义 BuildGraph 任务。

BuildGraph 是一个基于脚本的构建自动化系统，具有虚幻引擎 4 (UE4) 项目通用的构建块图。 BuildGraph 与 UnrealBuildTool , AutomationTool 和编辑器集成，并且可以针对您的项目进行扩展和定制。

图 2.1 –来自 UE4 文档

详细介绍 BuildGraph 的所有功能超出了本白皮书的范围。如果您想扩展构建系统，那么我强烈建议您从 [Build Graph Documentation](#) 开始

文档并真正地消化脚本和代码。幸运的是，Epic 包含一个预定义的自定义脚本，用于进行安装构建。您可以在以下位置找到此脚本：

[EngineDrop] /Engine/Build/InstalledEngineBuild.xml

该脚本是您安装构建的网关，并以标准 XML 标头开头。接下来的 100 行左右是我们将用来配置构建的各种选项。我将简要介绍如何利用这些选项。脚本的真实内容可在 <Agent> 及其 <Node> 标记中找到。这些定义了脚本将要采取的动作类型。同样，如何自定义这些脚本超出了本文档的范围，但是我们对三个不同的节点非常感兴趣：

1. <Node Name = “ Make Installed Build Win64” ...>构建一个 Win64 自定义安装版本。
2. <Node Name = “ Make Installed Build Mac” ...>构建 Mac 自定义安装版本。
3. Node Name = “ Make Installed Build Linux” ...>构建 Linux 自定义安装版本。

执行 BuildGraph 时，将在 **-TARGET=** 参数中使用与所需平台匹配的节点名称。

有两种方法可以运行 BuildGraph？首先，您可以在

[EngineDrop] / Engine / Build / BatchFiles 目录（可随时将其添加到您的路径中，因为它使工作更轻松）。这是执行自动化工具的首选方法，因为它会执行所有先决条件检查，甚至在需要时进行编译（并且存在 Visual Studio）。您也可以单独运行自动化工具。我们将使用 **RunUAT.bat** 方法。

我们需要讨论的最后一件事是可用的选项。这些是通过 **InstallEngineBuild.xml** 脚本中的 <Option> 标记定义的。如果愿意，可以直接在 xml 中设置它们，也可以使用 **-set**：

[option] = <value>格式通过命令行传递它们。在 **InstallEngineBuild.xml** 脚本本身中可以找到每个命令的详细说明。

对于大多数开发人员来说，Epic 为其 **InstallEngineBuild.xml** 脚本选择的平台选项的默认值是过高的。包含 Lumin 平台 (Magic Leap) 会引起实际问题。大多数开发人员不仅可以按原样构建安装版本，还可以编辑 xml 脚本，或者知道他们必须通过添加来禁用 Lumin suport，

“ -Set : WithLumin : false ”

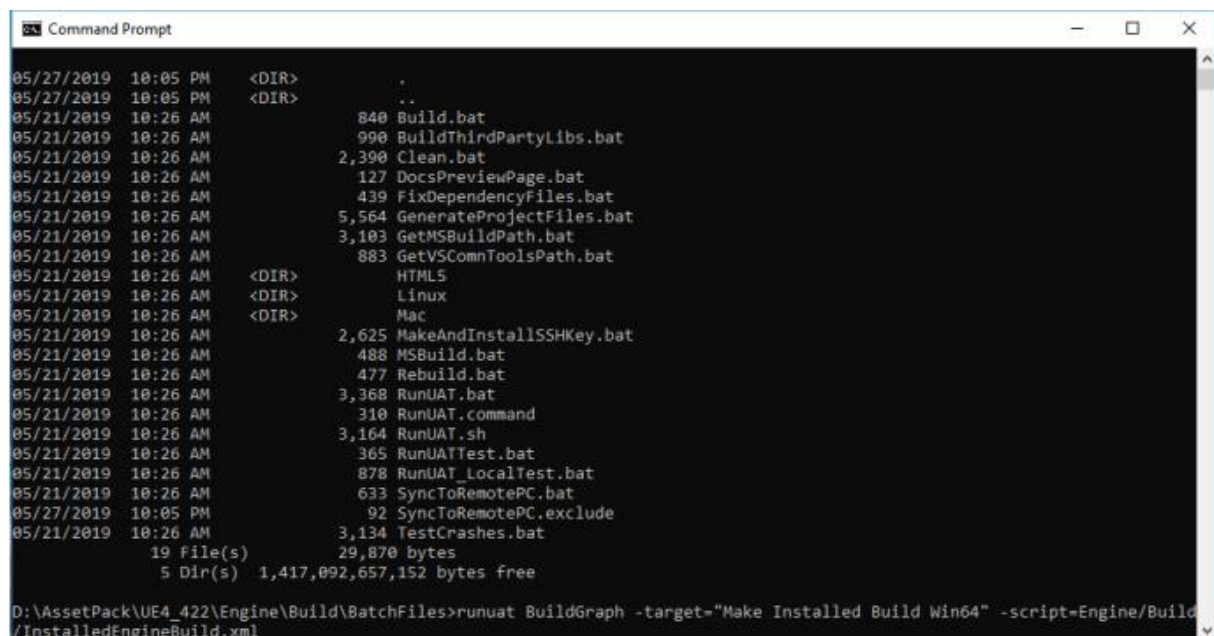
在命令行上，我建议始终检查 xml 脚本并为每个新引擎修复默认值，因为总是会添加新硬件。现在是时候继续执行我们的自定义安装版本了。

第 6 步—构建我们的 Development Drop

编辑 **[EngineDrop] / Engine / Build / InstalledEngineBuild.xml** 并根据需要进行配置。就我而言 (为了节省构建时间)，我将所有平台设置为 false，除了 **Win64** 以外。请记住，启用的平台越多，此步骤就越长。

配置就绪后，打开命令提示符并浏览以下目录：

[EngineDrop] / Engine / Build / Batchfiles



```
Command Prompt
05/27/2019 10:05 PM <DIR> .
05/27/2019 10:05 PM <DIR> ..
05/21/2019 10:26 AM      840 Build.bat
05/21/2019 10:26 AM     990 BuildThirdPartyLibs.bat
05/21/2019 10:26 AM    2,390 Clean.bat
05/21/2019 10:26 AM     127 DocsPreviewPage.bat
05/21/2019 10:26 AM     439 FixDependencyFiles.bat
05/21/2019 10:26 AM    5,564 GenerateProjectFiles.bat
05/21/2019 10:26 AM    3,183 GetMSBuildPath.bat
05/21/2019 10:26 AM     883 GetVSComnToolsPath.bat
05/21/2019 10:26 AM <DIR> HTML5
05/21/2019 10:26 AM <DIR> Linux
05/21/2019 10:26 AM <DIR> Mac
05/21/2019 10:26 AM    2,625 MakeAndInstallSSHKey.bat
05/21/2019 10:26 AM     488 MSBuild.bat
05/21/2019 10:26 AM     477 Rebuild.bat
05/21/2019 10:26 AM    3,368 RunUAT.bat
05/21/2019 10:26 AM     310 RunUAT.command
05/21/2019 10:26 AM    3,164 RunUAT.sh
05/21/2019 10:26 AM     365 RunUATTest.bat
05/21/2019 10:26 AM     878 RunUAT_LocalTest.bat
05/21/2019 10:26 AM     633 SyncToRemotePC.bat
05/27/2019 10:05 PM      92 SyncToRemotePC.exclude
05/21/2019 10:26 AM    3,134 TestCrashes.bat
      19 File(s)          29,870 bytes
       5 Dir(s)  1,417,092,657,152 bytes free

D:\AssetPack\UE4_422\Engine\Build\BatchFiles>runuat BuildGraph -target="Make Installed Build Win64" -script=Engine/Build/InstalledEngineBuild.xml
```

图 2.2

在图 2.2 中，您可以看到自定义安装版本所需的典型命令行。这个命令行假定您正在使用脚本本来配置要构建的平台。

按 Enter 键，然后去吃晚饭。进行自定义安装版本需要花费时间，时间的长短取决于所包含的平台数量。但是即使是构建 Win64 / Win32，在一台快速的 PC 上也可能需要一个多小时的时间。

那么认真，喝杯咖啡！

你觉得 Development Drop 怎么样？

既然您已经完成了自定义安装版本的构建，那么在开发过程中如何使用它？好吧，这几乎就像复制某些文件一样简单...差不多。在此之前，我们先讨论两件事，即 UE4 如何识别引擎的不同版本以及如何设置先决条件 (setup prerequisites)。

.UPROJECT ASSOCIATIONS

由于 Windows10 是我的主要开发平台，因此以下技术信息是：仅针对该平台。Mac / Linux 用户可能具有不同的机制。

UE4 在 Windows 中注册 **UPROJECT** 文件，并将其传递给 **UnrealVersionSelection.exe**。

这是由 Epic Games 开发的自定义实用程序，它将打开 uproject 文件并确定要使用的引擎版本。但是怎么知道呢？

每个 **UPROJECT** 文件都有一个选项 **EngineAssociation**。该选项告诉

UnrealVersionSelection 要使用哪种引擎版本。整个过程依赖于 Windows 注册表中的两个特定分支。

第一个分支位于 **HKEY_LOCAL_MACHINE \ SOFTWARE \ EpicGames \ Unreal**

Engine，此分支描述了所有 Epic 官方启动器版本。如果您通过 Epic Games 安装了该引擎的版本商店 (EGS)，有关此构建的重要信息将在上述密钥下找到。

第二个分支位于 **HKEY_CURRENT_USER \ Software \ Epic Games \ Unreal Engine **

Builds，可以在此处找到任何自定义安装版本。必须先在操作系统中注册，然后才能使用自定义安装版本。这是我将要介绍的先决条件步骤的一部分。

那么，该流程如何运作呢？双击 **uproject** 文件时，Windows 将执行

UnrealVersionSelection 的版本，并将 **UPROJECT** 文件作为参数传递。哪个版本由典型的

Windows 文件关联规则确定，并且密钥可以在 **HKEY_CLASSES_ROOT **

Unreal.ProjectFile 中找到。

UVS 将打开 uproject 文件并读取 EngineAssociation 值。它首先查看引擎的正式安装版本，以查看是否存在匹配项，如果存在，则加载该版本。接下来，它查看自定义安装的版本。如果在任何一个分支中均未找到匹配项，UVS 将打开一个对话框，允许您将 **uproject** 文件分配给现有引擎。

注意：当 Epic 将自定义安装的版本与 uproject 关联时，它将使用标准格式的 GUID。GUID 是唯一的标识符（由操作系统保证），因此很有意义，但您不必使用它。UE4 执行标准的字符串/文本比较，以查找要使用的内部版本！因此，我建议改用有意义的值！

行了 **setup.bat** 来安装先决条件。但是 development drops 呢？这是一个混合包。UE4 先决条件重做器包含在安装版本中，但不包含 **GITDependencies**，**UnrealVersionSelection** 和 **Setup.xxx** 文件。因此，为了创建一个完全可安装的版本，您需要手动复制一些文件。

```
[EngineDrop] /Setup.bat
```

```
[EngineDrop] /Engine/Binaries/Win64/UnrealVersionSelector-Win64-Shipping.exe
```

```
[EngineDrop] /Engine/Binaries/DotNET/GitDependencies.exe
```

```
[EngineDrop] /Engine/Binaries/DotNET/GitDependencies.exe.config
```

如果您使用的是 P4 或其他某些源代码控制系统，则 **GITDependencies** 是可选的，如果未安装 **setup.bat** 文件，则可以很好地跳过它。将这些内容复制到位后，您现在将拥有一个功能齐全的开发分支，可以将其打包以分发为您认为合适的对象。

步骤 7 –安装开发分支

现在是时候应对新的开发问题了。在 WisE，我们利用 perforce 分配 **Grop**，使艺术家可以轻松达到最高水平。我们只需签入开发位置，并告诉用户何时需要执行 **setup.bat**。

分发和安装 Drop 的方式完全取决于您和您的需求。只要记住，每次引擎更新时，**setup.bat** 至少需要运行一次。

附录 A –自定义工具

现在您已设置为使用自己的自定义安装版本，WisE 为您提供了两个工具。这些工具将使管理

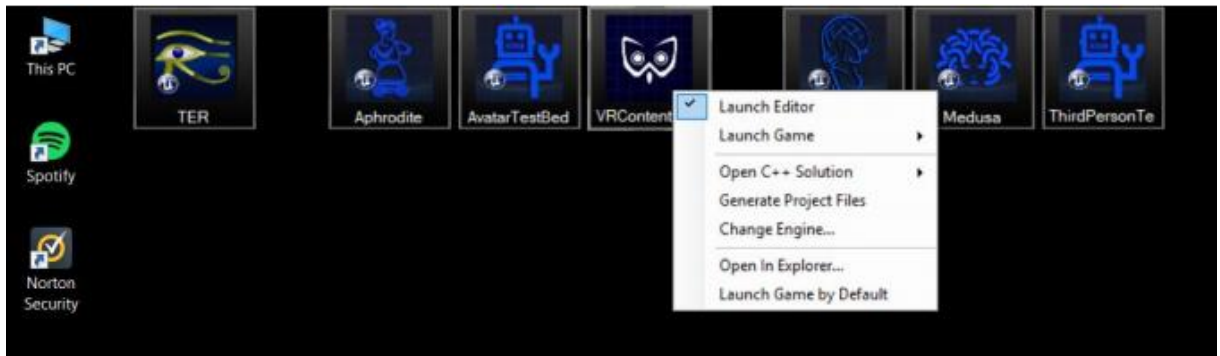
多个 UE4 项目变得更加容易。

UE4 注册

我们拥有的第一个工具是 **UE4Register**。手动编辑注册表可能很麻烦，而 **UE4Register** 则将其删除。它允许您一次快速注册引擎的多个版本。当您执行它时，它将查看其下面的所有子文件夹并注册找到的引擎的任何副本。当然，您可以在引擎目录中包含一个 **UE4Register.key** 文本文件，以定义要在其下注册引擎的密钥。

UE4 桌面

我们的第二个工具是 UE4Desktop。UE4Desktop 是一种轻量级的工具，可以在 Windows 桌面上创建类似于图标的桌面来管理 UE4 项目。



通过上下文菜单弹出窗口提供项目的常规操作。UE4Desktop 还支持打开项目的多个副本，并完全控制窗口的打开位置，并具有更多功能。您一定要检查一下。

有关 UE4Register 或 UE4Desktop 的更多信息，请查看我们的工具页面，网址为：

[WisE 工程资源页面](#)

附录 B – 常见问题解答：

我将尝试通过回答问题来使本节保持最新状态。

问：当我尝试构建自定义安装版本时，出现错误消息“无法找到 PDBCOPY.EXE 安装”。

答：由于某些原因，这些工具未随 Visual Studio 一起预安装。您需要自行安装。只需在

[Windows 10 SDK 下载页面上运行调试器](#)